Math 164: Problem Set 4

Owen Jones

2/2/2024

6.18 Let $\tilde{x} = \underset{x \in \mathbb{R}}{\arg\min} f(x)$ where $f(x) = \sum_{i=1}^{n} (x_i - x)^2$. By the FONC, $f'(\tilde{x}) = 0 \Rightarrow 2\sum_{i=1}^{n} (x_i - \tilde{x}) = 0$. Thus, $\tilde{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

6.21 (a) Let $\mathbf{x}^* = \underset{x \in \mathbb{R}}{\arg\min} f(x)$ where $f(x) = \frac{\sqrt{1+x^2}}{v_1} + \frac{\sqrt{1+(d-x)^2}}{v_2}$.
By the FONC, $f'(\mathbf{x}^*) = 0 \Rightarrow f'(\mathbf{x}^*) = \frac{\mathbf{x}^*}{\sqrt{1+\mathbf{x}^{*2}}v_1} - \frac{d-\mathbf{x}^*}{\sqrt{1+(d-\mathbf{x}^*)^2}v_2} = 0$
$\Rightarrow \frac{\mathbf{x}^*}{\sqrt{1+\mathbf{x}^{*2}}v_1} = \frac{d-\mathbf{x}^*}{\sqrt{1+(d-\mathbf{x}^*)^2}v_2}$
$\Rightarrow \frac{v_1}{v_2} = \frac{\frac{\mathbf{x}^*}{\sqrt{1+\mathbf{x}^{*2}}}}{\frac{d-\mathbf{x}^*}{\sqrt{1+(d-\mathbf{x}^*)^2}}} = \frac{\sin\theta_1}{\sin\theta_2}$.

(b) $f''(x) = \frac{1}{(1+x^2)^{\frac{3}{2}}v_1} + \frac{1}{(1+(d-x)^2)^{\frac{3}{2}}v_2}$. Because $f''(x) > 0$ for all x, the SOSC is satisfied.

6.25 Suppose $\mathbf{d}$ is a feasible direction at $\mathbf{x} \in \mathbf{\Omega}$. It follows, there exists some $\alpha_0 > 0$ s.t $\forall \alpha \in [0, \alpha_0]$ $\mathbf{x} + \alpha\mathbf{d} \in \mathbf{\Omega}$. Therefore $\mathbf{A}(\mathbf{x}+\alpha\mathbf{d}) = \mathbf{A}\mathbf{x} + \alpha\mathbf{A}\mathbf{d} = \mathbf{b} + \alpha\mathbf{A}\mathbf{d} = \mathbf{b}$
$\Rightarrow \mathbf{A}\mathbf{d} = \mathbf{0}$.
Suppose $\mathbf{A}\mathbf{d} = \mathbf{0}$. Let $\mathbf{x} \in \mathbf{\Omega}$, and let $\alpha > 0$. It follows $\mathbf{A}(\mathbf{x} + \alpha\mathbf{d}) = \mathbf{A}\mathbf{x} + \alpha\mathbf{A}\mathbf{d} = \mathbf{b} + \alpha \cdot \mathbf{0} = \mathbf{b}$. Thus, $\mathbf{d}$ is a feasible direction at $\mathbf{x} \in \mathbf{\Omega}$.

6.26 Since $\mathbf{0}$ is a boundary point, we need to show there exists some feasible direction $\mathbf{d}$ s.t $\mathbf{d}\nabla f(\mathbf{0}) < \mathbf{0}$. $\mathbf{d} = [1,1]^\top$ is clearly a feasible direction because both components are greater than 0. Because $f(\mathbf{0}) \neq \mathbf{0}$, $\frac{\partial f}{\partial x_1}(\mathbf{0}) \leq 0$, and $\frac{\partial f}{\partial x_2}(\mathbf{0}) \leq 0$, at least one of $\frac{\partial f}{\partial x_1}(\mathbf{0}) < 0$ or $\frac{\partial f}{\partial x_2}(\mathbf{0}) < 0$ must be true. Thus, $\mathbf{d}\nabla f(\mathbf{0}) = \frac{\partial f}{\partial x_1}(\mathbf{0}) + \frac{\partial f}{\partial x_2}(\mathbf{0}) < 0$. Hence, $f(\mathbf{0})$ cannot be a local minimizer because it doesn't satisfy the FONC.

6.27 Because $f(\mathbf{x}) = \mathbf{c}^\top\mathbf{x} \Rightarrow \nabla f(\mathbf{x}) = \mathbf{c} \neq \mathbf{0}$. Thus, for any interior point, $\nabla f(\mathbf{x}) \neq \mathbf{0}$, so there can't be any interior points that satisfy the FONC. Hence, if there exists a local minimizer over the set $\mathbf{\Omega}$, it must lie on the boundary of $\mathbf{\Omega}$

7.2(d) $f'(x) = 2x - 4\sin(x), f''(x) = 2 - 4\cos(x) \Rightarrow x^{(k+1)} = x^{(k)} - \frac{x-2\sin(x)}{1-2\cos(x)}$ gives us our formula for Newton's Method.
$x^{(0)} = 1, \quad x^{(1)} = -7.47274, \quad x^{(2)} = 14.47852, \quad x^{(3)} = 6.93511$

In [11]:
```
import numpy as np
import math as m
from math import sin,cos
import pandas as pd
from matplotlib import pyplot as plt
```

In [2]:
```
def g(x):
    df=2*x-4*sin(x)
    ddf=2-4*cos(x)
    return x-df/ddf
```

In [3]:
```
f=lambda x:x**2+4*cos(x)
```

In [4]:
```
def Newton(x_0,tol):
    x=np.array([x_0])
    while abs(g(x[-1])-x[-1])>tol:
        x=np.append(x,g(x[-1]))
    return x
```

In [5]:
```
arr=Newton(1,10e-20)
```

In [6]:
```
vf=np.vectorize(f)
```

In [42]:
```
df=pd.DataFrame({'$x^{(k)}$':arr,'$f(x^{(k)})$':vf(arr),'$\epsilon$':np.insert(a
df.index.name="iteration number"
```

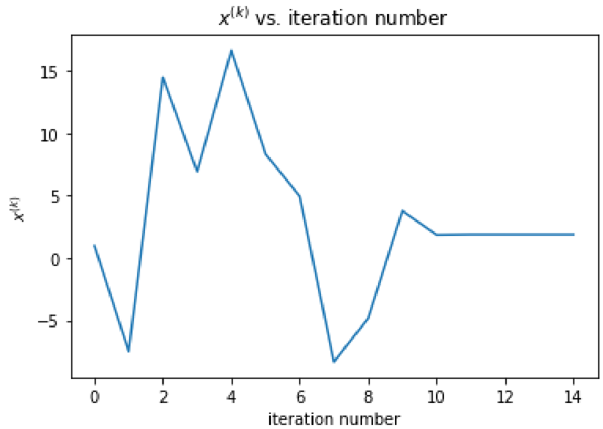
In [43]:
```
df
```

Out[43]:

iteration number	$x^{(k)}$	$f(x^{(k)})$	ϵ
0	1.000000	3.161209	NaN
1	-7.472741	57.330143	8.472741e+00
2	14.478521	208.288517	2.195126e+01
3	6.935115	51.275483	7.543406e+00
4	16.635684	274.347348	9.700569e+00
5	8.343938	67.738946	8.291747e+00
6	4.954633	25.507911	3.389305e+00
7	-8.301318	67.181618	1.325595e+01
8	-4.817320	23.625525	3.483998e+00
9	3.792574	11.201663	8.609894e+00
10	1.861061	2.318725	1.931513e+00
11	1.896214	2.316809	3.515312e-02
12	1.895495	2.316808	7.196385e-04
13	1.895494	2.316808	2.995885e-07
14	1.895494	2.316808	5.195844e-14

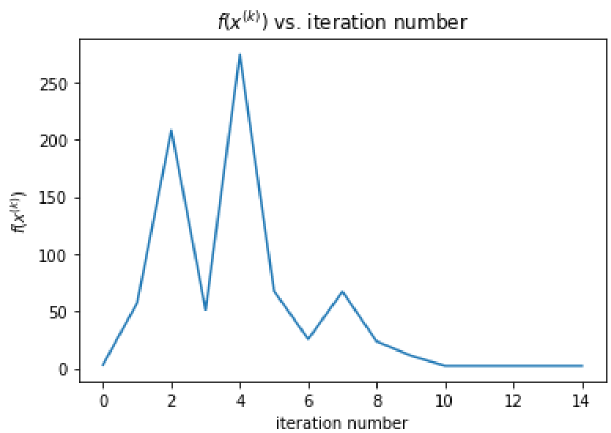
In [32]:
```
plt.plot(df.index,df.iloc[:,0])
plt.xlabel("iteration number")
plt.ylabel("$x^{(k)}$")
plt.title("$x^{(k)}$ vs. iteration number")
```

Out[32]:
```
Text(0.5, 1.0, '$x^{(k)}$ vs. iteration number')
```



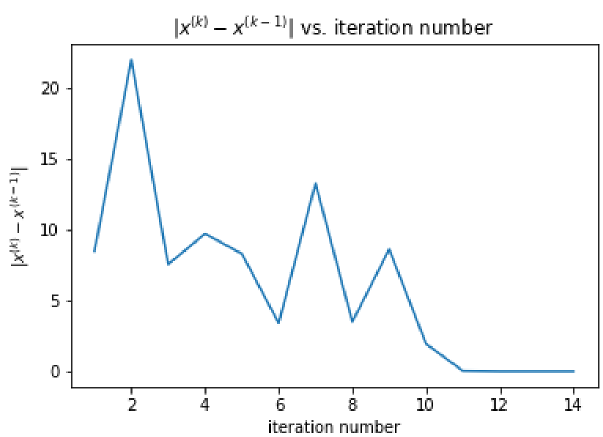
In [33]:
```
plt.plot(df.index,df.iloc[:,1])
plt.xlabel("iteration number")
plt.ylabel("$f(x^{(k)})$")
plt.title("$f(x^{(k)})$ vs. iteration number")
```

Out[33]: `Text(0.5, 1.0, '$f(x^{(k)})$ vs. iteration number')`



In [44]:
```
plt.plot(df.index,df.iloc[:,2])
plt.xlabel("iteration number")
plt.ylabel("$|x^{(k)}-x^{(k-1)}|$")
plt.title("$|x^{(k)}-x^{(k-1)}|$ vs. iteration number")
```

Out[44]: `Text(0.5, 1.0, '$|x^{(k)}-x^{(k-1)}|$ vs. iteration number')`



In [ ]: